Computer Vision & Image processing project

cap inspection and pre-processing

final report

A.Y. 2020/2021

Professor:

Di Stefano Luigi

Student

Wen Xiaowei

Matriculation N. 982501

Table of Contents

Introduction

For this project, I implemented a software to pre-process an image and get it ready for the OCR of the cavity number of a plastic cap.

The cap (1) has an external tab at a fixed position in relation to the cavity number (2).

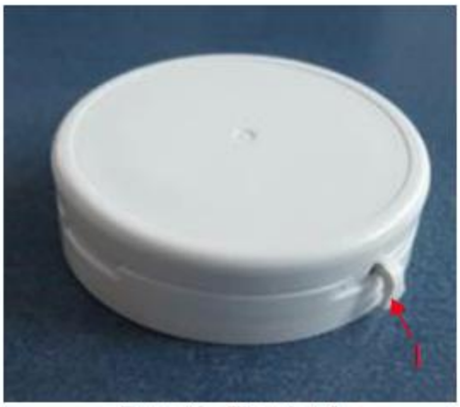


Figure 6 – The tab is 1.

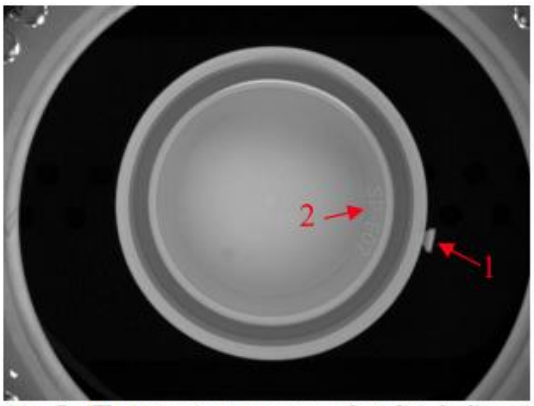


Figure 7 – The image to preprocess: the tab is 1, the cavity number is 2.

More detailed, the tasks are:

1. Outline the cap by generating a circle that fits the cap mouth.
2. Generate a crop containing the cavity number.
3. Generate a rectified crop containing the cavity number by applying a polar transform.

Notions Used

During the implementation of the project, following notions are used:

1. Filters (Gaussian Filter and mean filter).
2. Hough Transform for the circles.
3. Blob analysis (different properties are used to determine which one is the connected component which represents the tab).
4. Binary morphology.
5. Image warping (Polar transform).
6. Canny Edge Detection.
7. Hough-transform for the line.

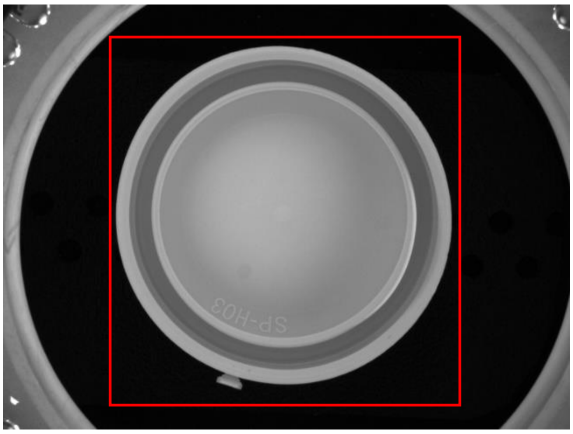
Libraries Used:

1. OpenCV2: this is the main library used to work with images.
2. Numpy: library used to manipulate images as arrays.
3. Matplotlib: used to plot the images.
4. Skimage.measure: used to do the blob analysis without implementing all the algorithms, contains the “label” method which allow to label different blobs, “regionprops” and “regionprops_table” are used to compute and visualize different properties of the blobs.
5. Skimage.transform.rotate: used to rotate the image.
6. Pandas: used to manipulate the table of properties of blobs.

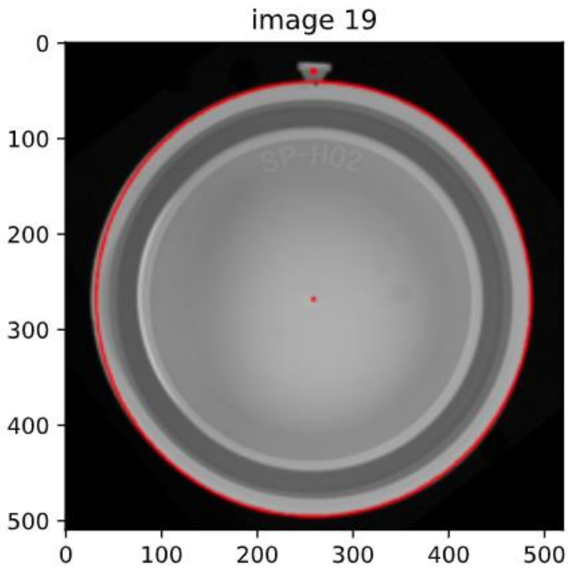
Stages

The project can be divided into different stages by splitting the main tasks in minor ones:

0. Importing the library and methods used.
1. Load the image, cropping out which are the part of the image that we need to work with:

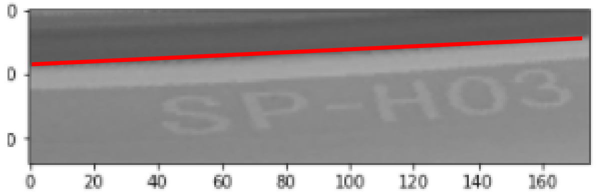


   We are only interested in the region inside the red square. Then I applied a Gaussian filtering to a copy of the image, because I will use the filtered one to compute the properties making the solution more robust, but as the result of this process will be used in a OCR, the images produces by this process will be made with the original unblurred one.
2. Defining 3 functions:
   a. find_best_circle(img): returns the best fitting circle to the cap.
   b. compute_centroid(img, area):
      given an image with a blob and the area of the blob it computes the centroid of the blob.
   c. compute_rotation_angle(delta_x,delta_y):
      given delta x and delta y, the difference of a point1 and point2, it computes how ma ny degrees we have to rotate to have the point on the y-axis of the cartesian plane with origin exactly on the point2.
3. In this step, I did different operations: in order to outline cap, I used the function defined at 2.3, then the circle is used also for the image segmentation leaving out only the tab if the cap is perfectly circular, but sometime this is not the case, so I had to computed several properties composing the table of properties, then, sorting based on compactness, area and rectangularity, I was able to decide which label the blob (tab) has got.
   Then, I can compute the centroid of the tab, the segment connecting the centroid and the centre of the circle, and finally, the angle between the segment and the Y-axis of the cartesian plan with the origin on the centre of the circle.

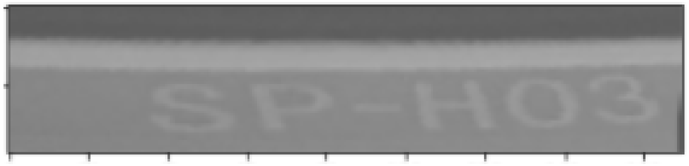


At this point, I rotated the image, saved the picture, in this way, the first task is complete.

4. In this step, I applied the polar transform and cropped the images in order to have only the region with the cavity numbers.
5. As the cavity numbers are rotated because of the polar transform, I had to apply a small rotation with an angle based the line highlighted in the figure below:



And obtaining this as the result:



6. Saving the images.

Final considerations

This project is very useful to use the theoretical notions learnt during the lectures and the labs.

As not all caps are perfectly circular, more operations are required to handle this problem: computation of the labels for all the blobs presenting in the image, their properties, I filtered many small blobs (sometime just pixels) to be able to choose correctly the blob representing the tab. And this operation is crucial because if we identify a blob as the tab, this can happen because when we segment the tap a portion of it can be left out, the successive operations as the rotation and so the crop of the cavity number will fail.

Websites

To complete this project, I visited following websites:

- https://docs.opencv.org/4.5.1/ opencv official documentation.
- https://scikit-image.org/docs/dev/api/api.html scikit-image official documentation.

- https://stackoverflow.com/ used to solve some problem during coding step.
- https://numpy.org/doc/stable/contents.html Numpy library's official documentation.
- https://www.geeksforgeeks.org/ tutorial of libraries

Bibliography

- Slides and notes of the lecture.